

μRobot Diego: Utilización de lógica borrosa en un robot rastreador

José Andrés Vicente, Belén Curto, Vidal Moreno, Javier Blanco

andres@tejo.usal.es, bcurto@abedul.usal.es, vmoreno@abedul.usal.es, jblanco@abedul.usal.es

Dpto. Informática y Automática. Universidad de Salamanca

Diego es un robot rastreador, capaz de seguir una línea negra sobre fondo blanco y de tomar la bifurcación correcta teniendo en cuenta las marcas dispuestas en el suelo. Para lograr este propósito se han incorporado técnicas provenientes del campo de la Inteligencia Artificial.

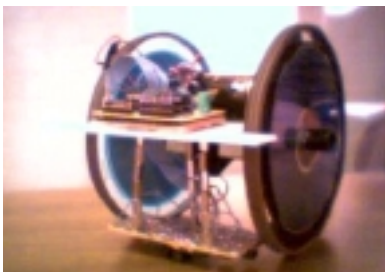


Fig. 1. Aspecto general de Diego

Descripción del hardware

En el diseño de Diego prima la sencillez y se ha buscado utilizar materiales de bajo coste.

Sistema sensorial

Su sistema de percepción consta de 8 sensores CNY-70 con una disposición en “V invertida” (Fig. 2).

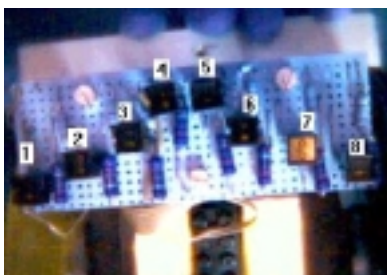


Fig. 2. Sistema sensorial

Actuadores

El sistema actuador consta de dos servos a los que se le han eliminado los topes y se le ha puesto una resistencia fija. Tienen una velocidad aproximada de 1 vuelta/s

Las ruedas se acoplan directamente sobre los servos en una configuración de giro diferencial. Con las grandes ruedas se alcanza una velocidad de unos 45 cm/s.

Para el accionamiento de los motores se generan señales PWM. Se han calculado los valores de ciclo PWM para el servo esté parado, gire hacia delante y hacia atrás. En la práctica se han considerado 8 niveles de potencia adelante y 8 atrás.

Procesador

Se utiliza el PIC 16F876 a 4 MHz. Dispone de 2 pines para generar PWM con una frecuencia de 250 Hz. Para la lectura de los sensores se utilizan el puerto B.

Alimentación

Se utilizan 4 pilas tamaño C para los servos y para los sensores (6 voltios). Se observó que utilizando pilas tamaño AA el robot tenía problemas de inestabilidad, por lo que se eligieron más grandes, que además aportan una mayor autonomía.

Para el PIC se utiliza un circuito 78L05 para reducir el voltaje a 5 V.

Técnicas y algoritmos

A partir de la información recogida por los sensores, el software debe cumplir dos funcionalidades: el seguimiento de la línea y el control de bifurcaciones. Para ello se ha utilizado el control borroso y se ha implementado una máquina de estados.

Como lenguaje de programación se ha elegido C, ya que facilita la labor y no existen situaciones críticas de tiempo o memoria.

Seguimiento de la línea

Para el seguimiento de la línea de la línea negra sobre fondo blanco, se han utilizado técnicas de control borroso, lo que permite suavizar el funcionamiento del robot. Se ha utilizado la dirección como variable de entrada con tres etiquetas lingüísticas (Fig. 3). La dirección se obtiene de buscar el primer bit a 1 por la derecha y sumarle el primer bit a 1 por la izquierda, lo que proporciona un intervalo continuo de valores entre 0 y 16.

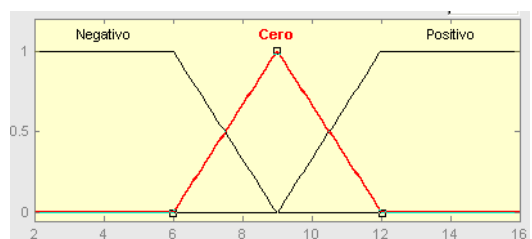


Fig. 3. Funciones de pertenencia de entrada

Se han definido dos variables de salida, una por cada motor. Cada variable tiene cinco etiquetas lingüísticas: negativo grande y medio, cero, y positivo medio y grande (Fig. 4).

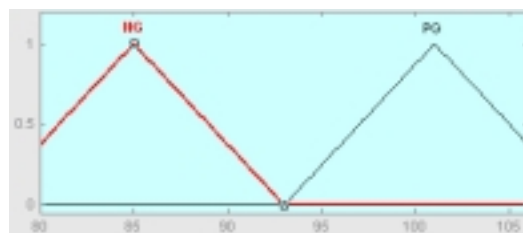


Fig. 4. Funciones de pertenencia para la salida

Las etiquetas son simétricas pero invertidas para cada motor, debido a que los motores están opuestos. Las reglas utilizadas son las siguientes:

Dirección	Negativa	Cero	Positiva
Motor izquierdo	PG	PG	NG
Motor derecho	NG	PG	PG

Tabla 1. Reglas borrosas

A partir de esta definición de reglas se calcula, fuera de línea, la tabla de potencias a aplicar a los motores utilizando herramientas de diseño. Para el control borroso únicamente se necesita incorporar esta tabla en el código C. Esto simplifica enormemente las necesidades de cálculo.

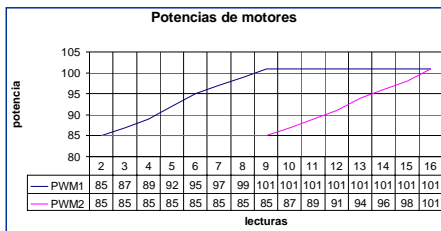


Fig. 5. Tabla de potencias

Detección de bifurcaciones

Sobre el circuito se encuentran dispuestas varias marcas, que definen el camino a seguir en la próxima bifurcación. La marca se identifica detectando huecos intermedios que se mantienen durante un tiempo determinado (Fig. 6). Para saber el lado donde se encuentra, se compara con la lectura justo antes de la marca (Fig. 7).

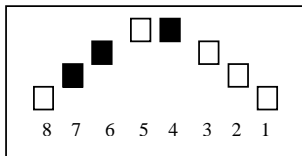


Fig. 6. Detección de una marca

Una vez determinado el lado, se hace control lateral para ignorar el camino incorrecto. El control lateral consiste en coger el primer bit a 1 por el lado de la marca y multiplicarlo por 2, así se obtendrán valores entre 0 y 16 al igual que el control por ambos lados.

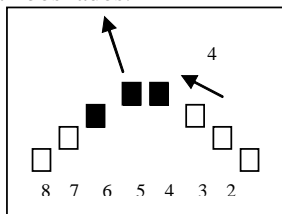


Fig. 7. Lectura previa

Estas situaciones se han formalizado en una máquina de estados, definida como es normal, por estados, eventos y acciones.

Estados

- 0 Control normal. Seguimiento de línea.
- 1 Detección de marca de bifurcación.
- 2 Control lateral según la marca leída.

Eventos

- A Hueco detectado
- A' Hueco durante un tiempo determinado
- B Línea detectada (sin huecos).
- C Temporizador. Se ha alcanzado un tiempo determinado.

Acciones

- En 0 Control normal.
- En 1 Control lateral opuesto a la marca.
- En 2 Control lateral del lado de la marca.

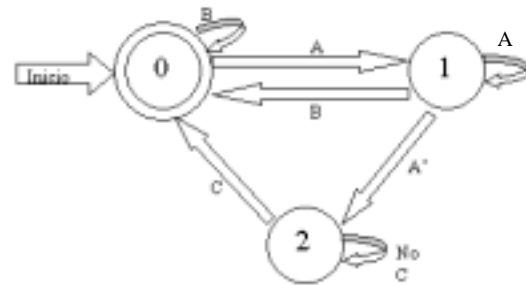


Fig. 8. Máquina de estados

Un ejemplo de los estados del robot ante una bifurcación es el siguiente:



Fig. 9. Ejemplo de transiciones de estados

Conclusiones

Este robot ha sido diseñado teniendo en cuenta las tolerancias de anchos de línea, longitud de marcas, separaciones, curvas, etc, según las normas para el concurso de rastreadores ALCABOT 2002.

Gracias a las técnicas de IA utilizadas, se ha alcanzado una enorme seguridad y suavidad en el seguimiento de líneas, y un nivel de detección de bifurcaciones correctas próximo al 100%.

La utilización de la lógica borrosa en el control de los motores, permite suavizar hasta niveles insospechados el funcionamiento del robot, permitiéndole trazar curvas de cualquier radio sin problemas. La curva de potencias obtenida con lógica *fuzzy*, sienta un precedente en el control de robots con giro diferencial.